associated privileges; policies to be enforced; and business or other processes, work flows, or other operations to be performed or initiated by instances of the smart container. In **302**, a smart container is instantiated using a smart container template. In some embodiments, a service running on content management server **106** of FIG. **1** responds to a user or other (e.g., implicit) indication that a smart container instance is required or desired by using template data to create and populate with initial attribute values and/or placeholder data or objects an instance of the smart container. In **304**, one or more policies are applied during runtime to managed content associated with or requested to be associated with a smart container.

[**0019**] FIG. **4** is a flow diagram illustrating an embodiment of a process for designing a smart container template. In some embodiments, **300** of FIG. **3** is implemented by FIG. **4**. In some embodiments, smart container models are saved as templates from which smart containers can be instantiated for run time. A smart container template may be created from scratch, an existing template or existing smart container instance. Upon saving a template configuration, a user interface provides a user options for organization of the template, overwriting existing templates, or saving as new. At the time of template creation, the user has the ability to specify if and how template changes will impact existing smart container instances based on the template as well as any child templates that were also based on the template. For example, the system can be configured, on a template by template basis, to propagate template changes to existing instances and/or child templates, propagate changes only if runtime instance policies match those of the source template (e.g., policies were not changed at instantiation or runtime), propagate changes regardless, or to not propagate changes. In some embodiments, a smart container template extends a given template, a parent template, in order to reuse common configurations. For example, a lending institution may create a Mortgage Loan File Smart Container Template, which contains the basic structure and policies required to gather and process documents for loan processing. However, different states may impose slightly different regulations. So a business analyst would like to create a California specific Mortgage Loan File template which extends the general template. In the example shown in FIG. **4**, in **400** membership is defined for a smart container template. In **402**, structure is defined for a smart container template. In **404**, policies are defined for a smart container template. In **406**, roles are defined for a smart container template. In some embodiments, one or more privileges are defined for each role.

[**0020**] FIG. **5** is a flow diagram illustrating an embodiment of a process for defining membership. In some embodiments, **400** of FIG. **4** is implemented by FIG. **5**. In the example shown, in **500** open or closed membership is specified. Open membership allows modification during runtime—for example, objects can be added during runtime. Closed membership does not allow modification during runtime—for example, a folder cannot be added to the smart container, or a document object not included originally in the composite object in the smart container cannot be added. An object can be a smart container (e.g., smart containers can be nested within each other), a folder, a category, a document, a query, a relationships, a placeholder object intended to be replaced by a corresponding instance-specific object at runtime, etc. In **502**, implicit or explicit membership is specified. Examples of explicit membership include folder containment, relationship (e.g., a finished goods specification for a product may reference individual components of material that are required to produce the specified product. For example, the individual components of material to package perfume may be bottle, cap, box, and label. These components of material may be represented as individual objects that are associated through a 'relationship' to one or more smart containers. It is by 'relationship' that the component is a member of a smart container), taxonomy (e.g., a taxonomy describes a hierarchical organization of a domain of knowledge—for example, a pharmaceutical taxonomy may organize information about different types of diseases and the drugs used to treat them as follows: a 'Pharma' smart container is organized in a structure represented as a folder structure with 'Pharma' containing a subfolder 'Diseases and Disorders,' which in turn contains a subfolder 'Blood Disorders,' which in turn contains a subfolder 'Anemia,' which in turn contains subfolders: 'Aplastic Anemia,''Dyserythropoietic Anemia,''Hemolytic Anemia, Megloblastic Anemia,' etc; the taxonomy structure gives folders membership in a taxonomy-based smart container), or category association (e.g., category association is related to taxonomies; and objects containing content and meta-data that match the concepts represented by folders in a taxonomy will be granted membership to those folders; in the taxonomy example above, if a document contains evidence that it is pertinent to aplastic anemia, then the document will become a member of the 'Pharma' smart container and linked to the 'Aplastic Anemia' sub-folder through category association.). An example of implicit membership is becoming a member by virtue of being returned in a query associated with or comprising the smart container. In some embodiments, each of the remaining steps may be applied/defined differently for the various nodes or objects of the smart container. In **504**, managed or unmanaged membership is specified. Managed members are governed by the policies associated with the smart container. Unmanaged members are included within the structure of the smart container but are not governed by the smart container's associated policies. In some embodiments, an implicit member (i.e. that of a query results), cannot be managed. In **506**, include in membership or exclude from membership rules are defined. A specific object can be defined such that it must be a member of the smart container (e.g., a loan application smart container must include a completed application form) or such that it cannot be a member (in which case an attempt to make an excluded object or an object of an excluded type a member will be rejected.

[**0021**] FIG. **6** is a flow diagram illustrating an embodiment of a process for defining structure. In some embodiments, process **402** of FIG. **4** is implemented by FIG. **6**. In the example shown, in **600** open or closed structure is specified. An open structure allows modification during runtime—for example, objects can be added during runtime. A closed structure does not allow modification during runtime—for example, a folder cannot be added to the smart container, or a document object not included originally in the composite object in the smart container cannot be added. In various embodiments, structure include one or more of the following as means to relate the individual objects that make up a compound object: 1) folder containment, 2) virtual documents, 3) relationships, 4) taxonomies, 5) categories (e.g. auto-categorization populates a specified attribute that